Learning saliency maps for object categorization

Franck Moosmann, Diane Larlus, Frédéric Jurie

Learning Saliency Maps for Object Categorization

Frank Moosmann, Diane Larlus, Frederic Jurie

INRIA Rhône-Alpes - GRAVIR - CNRS,
{Frank.Moosmann,Diane.Larlus,Frederic.Jurie}@inrialpes.de
http://lear.inrialpes.fr

Abstract. We present a novel approach for object category recognition that can find objects in challenging conditions using visual attention technique. It combines saliency maps very closely with the extraction of random subwindows for classification purposes. The maps are built online by the classifier while being used by it to classify the image. Saliency is therefore automatically suited to the abilities of the classifier and not an additional concept that is tried to fit into another method. Our results show that we can obtain state of the art results on commonly used datasets with using only little information and thus achieve high efficiency and short processing times.

1 Introduction

Object categorization has been heavily studied in the last decade, and a lot of breakthroughs have been made [1–7]. Most of these methods rely on local features used to represent local appearances of images. They can be detected by interest point detectors ([7–10]) chosen for their repeatability and their invariant properties to some transformations or randomly sampled from images [6]. Whatever strategy is used, the goal is to reduce the large amount of information carried by images and to focus on most relevant parts.
On the other hand, biologically plausible models of visual saliency have extensively been studied [11] but generally with low connection to complex recognition problems. Apart from the work of Walther *et al.* [12], who demonstrate the Koch and Ullman [13] saliency-based attention model can improve object recognition performances, most of the proposed works have only been used in simple applications.
When we come to visual search (or visual attention), even less work has been published showing its usefulness for complex object categorization applications. Visual search or visual attention is the process of selecting information based on saliency (bottom-up process) as well as on prior knowledge about objects (top-down process) [14, 15]. For the human visual system it seems to be an important component. However, biologically inspired systems mixing bottom-up and top-down processes, as far as we know, have never been able to perform better in terms of classification error than purely bottom-up approaches when dealing with complex recognition problems.

The main contribution of this article is to show a way to combine bottom-up and top-down processes in such a way that classification errors are much lower than using the bottom-up process alone. In our approach, the bottom-up process uses salient information, defined as the set of attributes which distinguish a concept (object category) the most from others. It is application/domain dependent and should be learned to fit with a given context. The top-down process relies on an online estimation of a probability density function, which estimates the probability of having an object part at a given position/scale in the image.

For this purpose, we propose a novel classifier which combines saliency maps with an object part classifier: prior knowledge stored by the classifier is used to simultaneously build the saliency map online as well as to provide information about the object class.

We measure the efficiency of the proposed method as the amount of information extracted from the image versus the classification performance. The method we propose produces results at a state of the art level using very small amount of information. It has been tested on several datasets, including the challenging GRAZ02 dataset[16].

After a brief review of previous related works (section 1.1), section 2 presents the tree based algorithm used for classification. Section 3 then explains how object-specific saliency information is used for building a visual attention map and how this is combined with the classifier. At last we present experiments in section 4, and come to the conclusions in section 5.

1.1 Related Work

Object class recognition as well as the detection of salient information are well studied problems and many different approaches have been proposed. This section will cover three related areas, i.e. object categorization, visual saliency and visual search, and will try to emphasize connections between them through the description of most noticeable works and concepts.

Object categorization. Object categorization can be defined as the capability to predict whether an object of a given category is visible or not. Proposed methods generally rely on a visual dictionary (called a *codebook*) learned from a set of training images. For example Csurka *et al.* [1] first detect sets of keypoints in training images, then represent local information in a neighborhood of these keypoints using SIFT descriptors [7] and finally produce a visual dictionary by clustering these descriptors with k-means. A similar approach is used by Fergus *et al.* in [2]. However, keypoints detectors are considered sometimes to be not repeatable enough in case of object categories which have large intra class variability. Winn *et al.* in [3] process every pixel, avoiding early removal of potentially useful regions such as textureless regions; their thesis advocate for not using any saliency based detector as it potentially discards good information.

Once local image representations are extracted and the visual dictionary computed, many approaches propose to use histograms of visual words to classify images [1–3, 17]. This approach is denoted as *bag-of-features* by an analogy to

learning methods using the bag-of-words representation for text categorization. Many other approaches rely on geometric models, such as the constellation model [18], fragment based models [5] or rigid geometric models [4].

Object categorization can also be addressed by classifying randomly selected images patches, i.e. determine if they can be a local part of any possible object category, and voting for the most frequent category [6].

None of these methods use any actual visual search strategy[1]: image features are extracted once at the beginning of the process and the way these features are selected comes from purely bottom-up mechanisms.

Visual Saliency. Even if visual search is not commonly used for categorizing objects, most advanced systems heavily rely on the extraction of salient features.

We can mainly distinguish four types of salient features used in literature. Keypoint based detectors [19, 7], generally based on cornerness criteria, can be defined to be scale, rotation, translation and even affine invariant [10]. As they allow to summarize images with a very small number of local information they make subsequent stages lighter. For this reason, it is very common to find them at the first stage of object categorization algorithms [1, 2, 4, 17].

Keypoints are very good at detecting similar local structures in a repeatable way but can't detect salient structures. Saliency should highlight what makes relevant information different from other ones. In [20] the saliency is defined as the sum of the absolute value of the local wavelet decomposition of the image. Kadir *et al.* [9] propose a measure relying on the entropy of distribution of local intensities.

However, it is clear that saliency does not always mean complexity, but is more related to the capability to determine which information most distinguishes a concept or an object from other possible concepts or objects. Walker *et al.* [21] define salient features as those having a low probability of being mis-classified with any other feature, while Vidal-Naquet *et al.* [5] propose to select image fragments which maximize the mutual information between the fragment and the object class. Fritz *et al.* [22] describe a system in which discriminative regions are produced by a conditional entropy measure. It has also being pointed out [23] that general low specific features are not as efficient as specific learned features, for recognizing objects. Saliency can also be defined as corresponding to the rarity of features. In [24] the saliency is defined over the probability in feature space, as being inversely proportional to the density.

All of these definitions are derived from man made criteria and are not biologically plausible. Models of biological vision have the appeal of its roots because biological systems are the only full-functioning systems known. Systems such as the one proposed by Itti *et al.* [11] has lead to interesting behaviors, but, as pointed out by Gao and Vasconcelos [25], the lack of a clearly stated optimality criteria for saliency constitutes a significant limitation of these methods. The approach proposed by Walther *et al.* [12] should be pointed out as they experimentally prove the validity of biologically inspired saliency-based models.

[1] also denoted as top-down strategies

Visual search and object recognition. Visual saliency and visual search are two related but different concepts. Exploring images through a sequence of fixations is supposed to make the interpretation task lighter by using top-down information. Despite the soundness of this concept, visual attention is not used in any of the most noticeable object categorization systems presented before.

However, systems based on visual attention have already been proposed in the past. Navalpakkam and Itti [26] propose to top-down-bias the visual search for detecting simple target objects, reducing by a factor of 2 the number of fixations. In [27], Bonaiuto and Itti show that a rapid pruning of the recognition search space improves the speed of object recognition. Avraham *et al.* [14] used a dynamic priority map for categorizing a list of regions.

Positioning of our approach. The approach proposed here can be considered as an extension of the method proposed in [6]. In that work, images patches are randomly selected and classified as belonging to one object category (background is considered as a category). We propose to bias this random selection by using a saliency map built online. We also improve the classification scheme by substituting the simple voting scheme with an SVM classifier.

2 Image classification

The framework we use to classify images is based on the work of Marée *et al.* [6] which consists of the following steps: first subwindows are randomly sampled on the training images, then randomized decision trees are built from these subwindows as a classifier. On the test images subwindows are again sampled randomly and each window is classified by the decision trees. In this section we will first describe the process of sampling subwindows and encoding the information, then describe the extremely randomized decision trees and finally we introduce a Support Vector Machine to replace the voting process and learn the importance of the leaf nodes of the decision trees.

2.1 Feature Extraction

To extract features from the image we do not use a detector function to localize salient parts of the image but instead sample subwindows at random positions with random sizes (of squared shape). Thereby a window must be fully contained in the image. In total we sample N_w windows which corresponds to N_i windows per image, depending on the amount of training images.

Each subwindow is then resized to 16x16 pixels and described by a descriptor function. Due to this resizing process we put the restriction on the sampling not to sample windows smaller than 16x16. Also, when sampling a large window and then resizing it to 16x16 pixel, much information will be lost, wherefore we introduce a parameter *MaxSize* (in percent of the image size) which can be specified by the user.

After resizing any descriptor function can be used to describe the subwindow. We tried different descriptors, with the result that the best choice is database

dependent: the Color-descriptor takes the pure color values in HSL color space and returns a 768-dimensional feature vector[2]. The Color-Wavelet-descriptor does a wavelet transform based on Haar basis functions [28] for each color channel and also returns a 768-dimensional feature vector. The last descriptor is the popular SIFT [7], which returns 4x4 histograms of 8 orientations. In the experiments we will state which descriptor we used.

2.2 Classification Trees

After feature extraction is finished we have to choose a learning method that can deal well with thousands of features in a high-dimensional space. Comparing the proposed EXTremely RAndomized Trees of Marée *et al.* [6] with Support Vector Machines yielded the result to continue using Extra-Trees. Learning is a lot faster and classification results were also better. This gives us a classifier that is able to classify single subwindows, not only whole images.

Extra-Trees are very similar to standard decision trees. Having to classify a feature consisting of n values (attributes), at every node one of those attributes is compared to a threshold, and comparison is continued either left or right until a leaf node is reached which holds the class label. The only difference between standard decision trees and Extra-Trees is that building the tree involves random decisions, which speed up the learning process enormously when having high dimensional data. A complete description of the algorithm can be found in [29].

Speeding up the learning procedure by introducing randomness also has some drawbacks. Compared to the standard decision tree learning the trees built are bigger and have higher variance. The first issue is not a problem because the trees are very fast for classification. The high variance on the other side decreases classification performance. [29] gives a good overview of methods to decrease variance in decision trees. One of the major possibilities is pruning, the other one is to build several trees and use the whole ensemble of trees to classify. Experiments showed when having several trees classification error reduces noticeably with an increasing number of trees, but since computation time also increases we decided on using an ensemble of 5 trees.

2.3 Learning the Importance of Tree Decisions

In order to classify an image subwindows are again sampled randomly and each window is then classified by the ensemble of decision trees. The votes of the trees are counted and the image classified as the class having most votes. In total we therefore get $noWindows * noTrees$ votes.

One property of our decision trees is that they fit the data perfectly. Other classification methods generalize better and have the ability to ignore unimportant data (e.g. SVM, K-means). Hence, leaf nodes of the tree that do not classify correctly influence the voting process equally to important leaf nodes. This is why we propose a method to learn the importance of the tree decisions.

[2] 16 pixel x 16 pixel x 3 color values

To learn this importance we use a binary vector whose size is equal to the total number of leaf nodes. At the beginning of the classification process of an image we set all entries of that vector to "0". When classifying a subwindow, we obtain a list of leaf nodes that are responsible for the votes. Having this list we set the corresponding entries in the vector to "1". Figure 1 illustrates this process. After all subwindows were classified, the entries being "1" in the vectors denote all leaf nodes that are responsible for the decision. When doing tests on

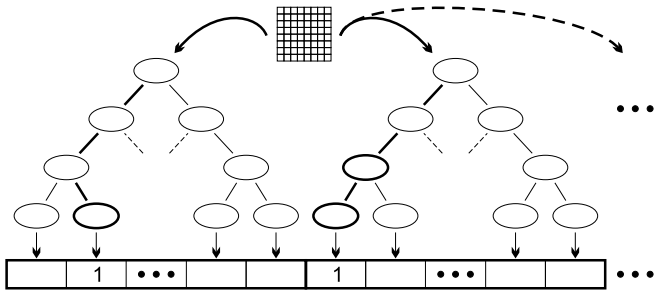

Fig. 1. Creating a feature vector when classifying with decision trees

the GRAZ-02 database we sampled 30000 subwindows in total on the training images and we obtain 5 trees that have approximately 4000 leaf nodes each. Assuming we sample 500 subwindows per test image, we get a 20000-dimensional "leaf-vector" of which at maximum 500 entries[3] are equal to "1".

Afterwards, a SVM is used to learn the importance of leaf nodes. To do this the classification process is applied to the training images. This gives us the same number of leaf-vectors as we have training images. With these vectors we now train a SVM with a linear kernel. To give the SVM classifier some more information, we increase the leaf-vector size by one and attach the category label that the voting returned.

When classifying an image, we sample random subwindows as described before, classify those windows with the decision trees and then classify the resulting leaf-vector with the SVM. The SVM-output is then our final decision on the category of the image. The experiments show that this method improves classification performance significantly.

In this section we have described the classification method we used and proposed some extensions on the works of Marée *et al.* that really improve the performance, as our experiments prove. Next we will show how to further improve results by incorporating saliency maps.

[3] some leafs might have been "hit" more than once

3 Learning saliency maps

In this section we introduce saliency maps that are adapted to the classifiers needs. First we are going to argue that saliency should be defined as the discriminativeness of features. We explain how we can bias the sampling of random subwindows with the help of saliency maps to increase classification performance. Then we show how these saliency maps can be created efficiently. Subsequently we demonstrate a way of building these maps online during the classification process.

3.1 Saliency map as a probability of finding objects of interest

Saliency maps contain information about where in the image interesting information can be found. These areas correspond to features considered as rare or informative, depending on the definition of saliency (see section 1.1). High saliency regions correspond to objects or places they are most likely to be found, while lower saliency is associated to background. Hence, this information can be used as a prior for a classification system to detect and classify objects. As showed in section 1.1, most approaches so far are based on a bottom-up process, independent from the task. We argue that it is more useful for a classification system to have prior knowledge about where the classifier can detect features that are discriminative to identify objects.

The classifier that we introduced extracts subwindows at random positions with random size and classifies each extracted subwindow. To improve the performance of the classification process we could use the prior information where objects are located and sample windows mainly at these locations. Now, sampling a subwindow by random corresponds to picking one point out of the three-dimensional scale-space (position x,y of the window and its size) according to some probability density function. Hence we define this 3-dimensional-PDF as the saliency map, in which a point is defined as salient, if the classifier will classify this point (subwindow) as non-background. For the classifier this is equivalent to the probability $P(O|X)$, where $X = (x, y, s)$ is the position and scale of having an object O at this point. This kind of idea was already mentioned in [30]. The described saliency map, that represents a PDF, can then be used as prior knowledge to bias the sampling of subwindows.

3.2 Building saliency maps by random sampling

To build the kind of saliency map that is suited to a classifier's ability the classifier itself is needed. Our method works in a way that first the classifier is built from the set of training images. Afterwards this classifier can be used to build the saliency map of any image. As stated, our classifier has the ability to classify each window, which means each point in the 3-dimensional scale-space. Using this ability we can generate a saliency map out of the results from classifying windows either as background (non-salient) or as objects (salient). To create this map efficiently we propose to randomly sample points in this space, then to classify the windows

using the classifier, and update the information of the saliency map depending on the output of the classifier. This will give us a number of "certain" points whose saliency values we know. From these points we can then propagate the saliency values to its neighbors and will finally obtain $\hat{P}(O)$. Methods exist to estimate distributed phenomenons like this offline, but instead we will focus on the online estimation of such a saliency map.

3.3 Active image classification with saliency maps

So far we introduced a definition of saliency maps and explained how they can be used to bias the sampling of subwindows. Now we show how these maps can be built online and used at the same time by the classification process.

To achieve the goal to guide the sampling in a way that enough windows are sampled on the object we introduce a probability density function for the sampling process that combines the probability for an object to be present at a given position and scale with the probability of having already explored a given area of the image space. This gives us the following probability density functions: $\hat{P}(O|X, Z_{1:n-1})$, the knowledge about having an object given the last $n-1$ measurements Z, is the saliency map we already talked about. Additionally we model the information about where windows already have been sampled with another PDF expressing the degree of need for exploration $P(E|X, S_{1:n-1})$ given the last $n-1$ samplings S. In order to sample windows we now have to combine these 2 informations to obtain a single PDF to draw random numbers from. This is done by multiplying the two PDFs which gives us the probability density function for sampling the next window

$$P(S_n = X) = \frac{\hat{P}(O|X, Z_{1:n-1}) \cdot P(E|X, S_{1:n-1})}{\sum_X \hat{P}(O|X, Z_{1:n-1}) \cdot P(E|X, S_{1:n-1})}$$

Patches which have a high probability of being on an object and which belong to a region of the image not explored yet will be selected with a higher probability.

To estimate $\hat{P}(O|X)$ we initialize our discrete PDF with a uniform distribution. For each window that we sample we adjust this distribution according to the classification outcome before we sample the next window. If a window was sampled at the point (x, y, s) and classified as object, we increase the PDF in its neighborhood $(x \pm c, y \pm c, s \pm c)$ by a constant amount and normalize the PDF afterwards. In the case it was classified as non-object we instead decrease the PDF in the same neighborhood. The radius of the neighborhood c that was used during our tests was 5% of the image size. Incrementing by a constant amount in a "cubic" region seems to be very simplified in a first view, but when doing this several times and adding up the numbers this produces smooth outcomes, as the test results show.

The need for exploration $P(E|X, S_{1:n-1})$ is similar to the estimation of $\hat{P}(O|X)$ but independent from the outcome of the classification. We also initialize it with a uniform distribution. Every time we draw from this distribution to sample a window reduces the need for exploration in this region. We therefore set the probability at this point to 0 and reduce the values of the neighbors in the same way

we increase/decrease $\hat{P}(O|X)$, with the difference of using the smaller radius of 3%.

This whole process can also be viewed from the perspective of online-filtering [31]. Our approach corresponds to the grid based method which gives optimal state estimates. Since our image will not change in time we have the identity function as state evolution model why we can skip the prediction step and directly apply the weight update like we described before.

We showed in this section a sound definition of saliency based on the discriminativeness of features and how a saliency map can be built online during the classification process. In the next section we will show some results that demonstrate the improvements yielded by the methods we proposed.

4 Experiments

Our experiments aim to measure how the combination of the proposed bottom-up and top-down mechanisms improve performances for visual categorization applications. The final goal is to determine the category of images depending on objects they contain.

These experiments were carried out on different datasets: the "GRAZ-02" dataset introduced in [16], the dataset used in the Pascal challenge 2005 [32] and the horses dataset introduced in [33]. Because our method includes a high rate of randomness the results differ in each run; we ran tests several times averaged the results and computed the variance. We measure performances with ROC curves, summarized by the true positive rate at the equal error rate (EER) point, which is obtained when the number of true positive plus the number of false positive equals 1.

4.1 Experiments on "GRAZ-02"

The GRAZ-02 dataset contains four categories: bikes (B), cars (C), persons (P) and counterexamples (N, meaning that it contains no bikes, no persons and no cars). This database is challenging in a way that objects are not always fully visible, images are sometimes rotated, objects are shown from different perspectives, the scale of objects varies heavily and illumination is not constant. It is also balanced with respect to the background, so it is not possible to detect an object based on its context, e.g. cars by detecting a traffic sign. There are no images including objects of different categories, but sometimes several objects of the same category. The goal is to label the test images with one of the labels 'B','C','P' or 'N'.

For all tests we took 300 images from each category, images with even numbers were used for training, images with odd numbers for testing. Several tests were carried out: in setting 1 we did not use the segmentation masks and train on the whole image, which is also the setting [16] used. In setting 2 we used the provided segmentation masks and trained only on the object itself. In both settings only

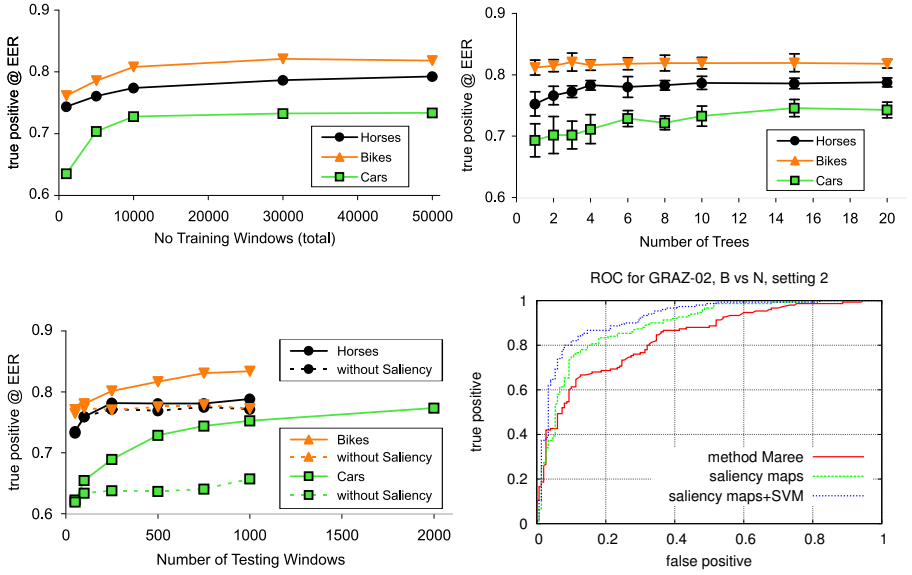

Fig. 2. Evaluation of parameters on different datasets. Brackets indicate the deviation

one of the object categories were tested against counter samples. We decided to do extensive tests on the two hardest categories, bikes and cars. As a descriptor the Wavelet Transform of the HSL color space was used for a maximum window size of 15%. Together with the "horse" database (described below) we evaluated some settings of our method that can be seen in figure 2.

With increasing number of sampled subwindows in the training images the classification rate also increases. Interesting are these results if we look at the absolute numbers: with 100 windows per image[4] the results already stabilize and do not improve much more.

For the number of decision trees one can also observe "the more the better". One can clearly see that this variance reduction technique works well in 2 ways: the more trees we have, the lower the variance is and the higher the classification rate is. Both parameters unfortunately also have in common that calculation time increases.

The third graph shows how the classification rate increases when increasing the number of subwindows sampled on test images. The dotted lines here indicate the tests without saliency maps. Hence saliency maps really push results when enough windows are sampled. The reason is that they are built online and therefore need some time to become accurate enough to push results.

The results shown in these three graphs all do not use the SVM classifier which improves results even further. This can be seen in the last graph, the ROC-curves for a test with setting 2 made on the GRAZ-02 database. It shows the results of our method with and without SVM compared to the same test taken with the

[4] GRAZ-02 included 300 training images, the "horse" database 100

method proposed by Marée *et al.* [6]. Even though we used only 5 trees and 30000 training windows (100 per image), compared to 10 and 100000 that Marée used, we achieve a much higher performance. The overall performance of the tests is listed in table 1. It can be further improved by using more than 500 subwindows for testing. However, this increases the calculation time. To obtain the stated results our method needed only 1-2s per image!

Table 1. Classification rate (at EER) for the GRAZ-02 database

	setting 1		setting 2	
	B vs. N	C vs. N	B vs. N	C vs. N
Result of Opelt *et al.* [16]	0.765	0.707	-	-
Method Marée *et al.*	-	-	0.736	0.621
With saliency maps	0.75	0.663	0.821	0.728
Saliency maps + SVM	0.799	0.717	0.855	0.83

To illustrate the test outcomes, we selected some images from tests done with setting 2 which are shown in figure 3. Each image is shown together with the resulting saliency map.

To demonstrate the ability of our method to generalize well, we also tried training on all four categories and testing additionally on different images taken from the internet (setting 3). This test included the use of segmentation masks for training. Figure 4 shows some results for this test setting. In the leftmost image it can be seen that it detects bikes and even trucks, which we did not train on. The method even generalizes well in images that do not contain any of the objects we trained on and correctly identifies unimportant background that it learned. To achieve these results of setting 3 we used 60000 training subwindows in total (100 per image).

4.2 Experiments on the Pascal Challenge dataset

To compare the classification performance of our method with others, we also did some tests according to the classification competition of the Pascal Challenge 2005. The dataset is a mixture of existing datasets and contains the four categories motorbikes, bicycles, people and cars. The goal is to distinguish one category from the others. We carried out tests on the test setting 1 and compared them to the results of the challenge. Only 73 subwindows per image were used for training a with maximum size of 30%, the SIFT descriptor was used to encode the patches and 4 Extra-Trees were used for classification. 10000 subwindows were extracted to feed the SVM. The results are in average 0.958 on motorbikes (1.1), 0.901 on bicycles (1.2), 0.94 on people (1.3) and 0.96 on the last category, the cars (1.4). These results are comparable with the best methods competing in the Pascal Challenge[5] but in contrast to them, we are using less information

[5] Taking only our best run we outperform the winning method in 2 of the 4 categories

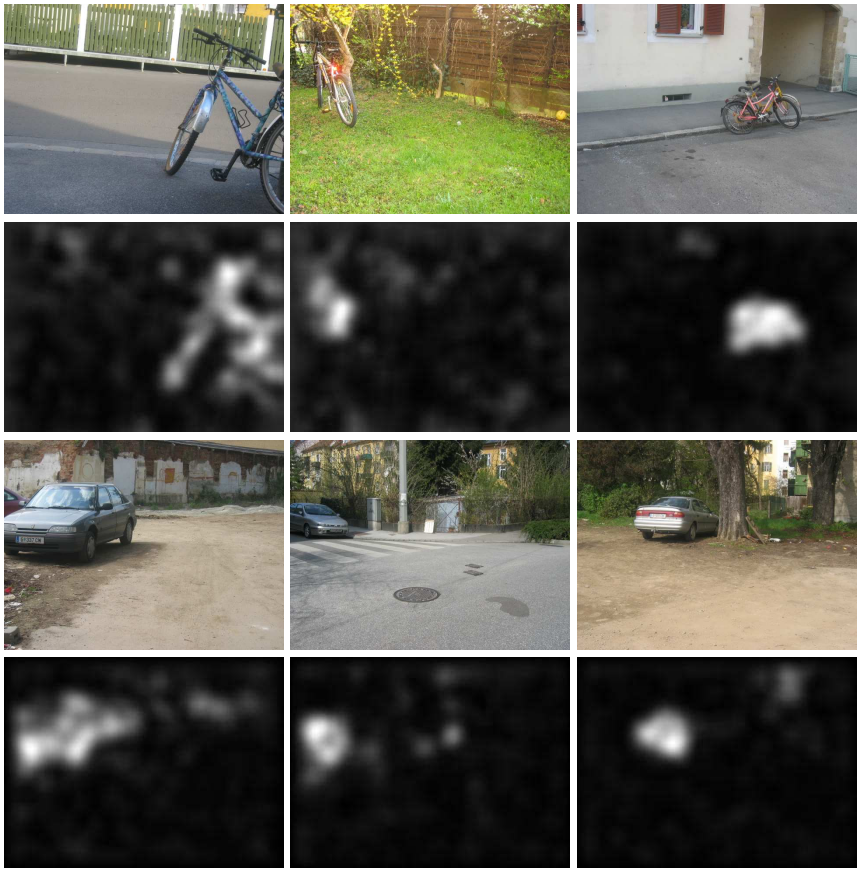

Fig. 3. Resulting Saliency Maps for the GRAZ-02 database, setting 2
Each image is shown together with the resulting saliency map below

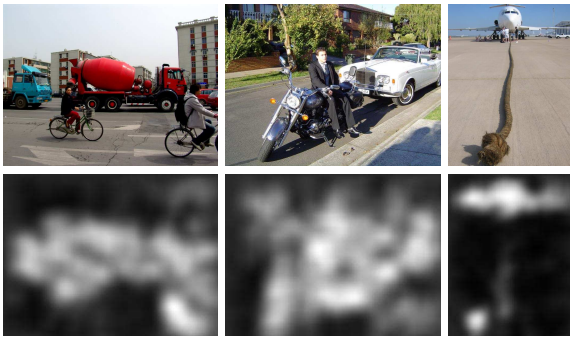

Fig. 4. Resulting Saliency Maps for images from google, training on GRAZ-02 images
Each image is shown together with the resulting saliency map

and have faster processing times. The execution times on a P4-2.8GHz were in average around 12 hours. This is substantially faster then the winning methods of the challenge.

4.3 Experiments on the Horses Database

This database was introduced in [33], but unfortunately got corrupted. It was recreated and published[6] with similar images by Jurie and Ferrari and contains 2 categories: horses and other images (non-horse images). It is difficult in a way that the images are taken randomly from the internet and thus are not biased in any way. Horses can be small, in various poses and can be occluded or only sketched. By using the SIFT descriptor we get a classification rate of 0.853 at the EER point, which can be considered a very good result.

5 Conclusions

In this paper we showed an efficient approach for object categorization. By defining saliency as the ability of a classifier to distinguish features well, building these saliency maps online and biasing the random sampling of subwindows we were able to really improve results compared to just uniformly sampling windows. This resulted in a very efficient classification system that can process images in less than 2 seconds. Still, our results are comparable with state-of-the-art methods which in contrast use much more information.

Nevertheless we believe that this method can be even further improved. Future work will contain trying to use the output of the SVM-classifier to update the PDF in a more intelligent way. It will also be interesting to try this method on datasets with more classes. Furthermore this method can be extended to not only categorize an image, but detect/localize objects.

References

1. Csurka, G., Dance, C., Fan, L., Williamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: ECCV'04 workshop on Statistical Learning in Computer Vision. (2004) 59–74
2. Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: Learning object categories from google's image search. In: ICCV. (2005) II: 1816–1823
3. Winn, J., Criminisi, A., Minka, T.: Object categorization by learned universal visual dictionary. In: ICCV. (2005) II: 1800–1807
4. Leibe, B., Schiele, B.: Interleaved object categorization and segmentation. In: BMVC. (2003)
5. Vidal-Naquet, M., Ullman, S.: Object recognition with informative features and linear classification. In: ICCV. (2003) 281–288
6. Marée, R., Geurts, P., Piater, J., Wehenkel, L.: Random subwindows for robust image classification. In: CVPR 2005. Volume 1. (2005) 34–40

[6] http://pascal.inrialpes.fr/data/horses

7. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV **60** (2004)
8. Harris, C., Stephens, M.: A combined corner and edge detector. In: AVC. (1988)
9. Kadir, T., Brady, M.: Saliency, scale and image description. IJCV **45** (2001)
10. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. IJCV **60** (2004) 63–86
11. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. PAMI **20** (1998) 1254–1259
12. Walther, D., Rutishauser, U., Koch, C., Perona, P.: On the usefulness of attention for object recognition. ECCV (2004)
13. Koch, C., Ullman, S.: Shifts in selective visual attention: Towards the underlying neural circuitry. Human Neurobiology **4** (1985) 219–227
14. Avraham, T., Lindenbaum, M.: Dynamic visual search using inner-scene similarity: Algorithms and inherent limitations. In: ECCV. (2004)
15. Zaharescu, A., Rothenstein, A.L., Tsotsos, J.K.: Towards a biologically plausible active visual search model. In: WAPCV. (2004) 133–147
16. Opelt, A., Pinz, A.: Object localization with boosting and weak supervision for generic object recognition. In: SCIA. (2005)
17. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. IJCV **43** (2001) 29–44
18. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR. (2003) II: 264–271
19. Schmid, C., Mohr, R., Bauckhage, C.: Evaluation of interest point detectors. IJCV **37** (2000) 151–172
20. Sebe, N., Lew, M.: Comparing salient point detectors. PRL **24** (2003) 89–96
21. Walker, K., Cootes, T., Taylor, C.: Locating salient object features. In: BMVC. (1998)
22. Fritz, G., Seifert, C., Paletta, L., Bischof, H.: Entropy based saliency maps for object recognition. In: ECOVISION. (2004)
23. Serre, T., Riesenhuber, M., Louie, J., Poggio, T.: On the role of object-specific features for real world object recognition in biological vision. In: BMCV. (2002)
24. Hall, D., Leibe, B., Schiele, B.: Saliency of interest points under scale changes. In: BMVC. (2002)
25. Gao, D., Vasconcelos, N.: Discriminant saliency for visual recognition from cluttered scenes. In: NIPS. (2004)
26. Navalpakkam, V., Itti, L.: Sharing resources: Buy attention, get recognition. In: WAPCV. (2003)
27. Bonaiuto, J., Itti, L.: Combining attention and recognition for rapid scene analysis. In: WAPCV. (2005)
28. Stollnitz, E.J., DeRose, T.D., Salesin, D.H.: Wavelets for computer graphics: A primer, part 1. IEEE Computer Graphics and Applications **15** (1995) 76–84
29. Geurts, P.: Contributions to decision tree induction: bias/variance tradeoff and time series classification. PhD thesis (2002)
30. Ye, Y., Tsotsos, J.K.: Where to look next in 3D object search. In: Proc. IEEE Int. Symp. Computer Vision. (1995)
31. Arulampalam, S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. IEEE Transactions on Signal Processing **50** (2002) 174–188
32. : The 2005 PASCAL Visual Object Classes Challenge. LNAI, Springer (2006)
33. Jurie, F., Schmid, C.: Scale-invariant shape features for recognition of object categories. In: CVPR. Volume II. (2004) 90–96